This paper was presented as part of the Mini-Conference at IEEE INFOCOM 2011

Describing and Forecasting Video Access Patterns

GONCA GÜRSUN MARK CROVELLA IBRAHIM MATTA
Department of Computer Science, Boston University

***Abstract*—Computer systems are increasingly driven by workloads that reflect large-scale social behavior, such as rapid changes in the popularity of media items like videos. Capacity planners and system designers must plan for rapid, massive changes in workloads when such social behavior is a factor. In this paper we make two contributions intended to assist in the design and provisioning of such systems. We analyze an extensive dataset consisting of the daily access counts of hundreds of thousands of YouTube videos. In this dataset, we find that there are two types of videos: those that show rapid changes in popularity, and those that are consistently popular over long time periods. We call these two types *rarely-accessed* and *frequently-accessed* videos, respectively. We observe that most of the videos in our data set clearly fall in one of these two types. In this work, we study the *frequently-accessed* videos by asking two questions: first, is there a relatively simple model that can describe its daily access patterns? And second, can we use this simple model to predict the number of accesses that a video will have in the near future, as a tool for capacity planning? To answer these questions we develop a framework for characterization and forecasting of access patterns. We show that for frequently-accessed videos, daily access patterns can be extracted via principal component analysis, and used efficiently for forecasting.**

I. INTRODUCTION

Video sharing is one of the most popular applications on the Internet. The largest video sharing site is YouTube, owned by Google Inc. According to [9], approximately 2 billion videos are watched and hundreds of thousands of new videos are uploaded every day. Today, Google generates $6-10\%$ of all Internet traffic and its largest contributor is YouTube [8]. This level of demand makes system design and capacity planning important issues for such sites.

Despite the importance of these issues, very little work has characterized the dynamics of individual video accesses over time. To help fill this gap, this paper makes two contributions. First, we characterize a workload that consists of user accesses to individual videos. Second, we show how to use these characterizations to predict future demand.

To do so, we analyze a dataset consisting of the daily time series of 100,000 YouTube videos. In this dataset, we find that there are two types of videos: those that show rapid changes in popularity, and those that are consistently popular over long time periods. We call these two types of videos *rarely-accessed* and *frequently-accessed* videos, respectively. We observe that most of the videos in our data set fall clearly into one of these two classes. In this work, using the *frequently-accessed* dataset, we study two questions: first, is there a relatively simple model that can describe the daily access patterns of *frequently-accessed* videos? And second, can we use this simple model to predict the number of accesses that a video will have in the near future as a tool for capacity planning? For the study on the *rarely-accessed* videos, we refer the reader to [7].

Our results show that there is a small set of common patterns that describe frequently-accessed videos. We also show how to leverage this small set of common patterns in order to predict future daily views for individual videos.

We show that common patterns can be extracted via principal component analysis. We show that approximately 20 principal components are sufficient to summarize the most popular 1000 videos. We then use these principal components in order to efficiently predict future daily views for each video using autoregressive models. In this way, we show how to efficiently forecast next-day access counts for with low absolute relative error.

The rest of the paper is organized as follows. We describe our dataset in Section II. We then introduce two key methods in Section III and present our main results in Section IV. In Section V we review related work and we summarize our contributions in Section VI.

II. DATASET

One of the strengths of our study derives from our dataset. We obtained it directly from Google, and it represents a global view of video accesses observed at YouTube servers. In contrast to datasets used in YouTube characterization to date, our data set is not restricted by video category (e.g. entertainment or sports) nor by the recommendation system of YouTube. The entire dataset is 326 GB in size and consists of millions of videos. From this large dataset, we select a subset consisting of the most popular 100,000 videos on April 1st, 2008. For each video, the available information is a one year long time series of daily views (from February 25th 2008 to February 25th 2009) and a unique identifier that does not reveal the video's actual name or category. Hence, no metadata on videos is available.

As already mentioned, we find two different behaviors in video access time series: some videos are consistently popular over long time periods, while the others show rapid changes in terms of popularity and are viewed only on a small number of days. Based on this observation we divide our dataset into two categories: *frequently-accessed videos* and *rarely-accessed videos*, respectively.

Figure 1 illustrates the number of days a video has at least one view. Based on Figure 1, we separate videos into two categories: those that have at least one access on more than half of the days in the year (frequently-accessed) and those that

978-1-4244-9921-2/11/$26.00 ©2011 IEEE

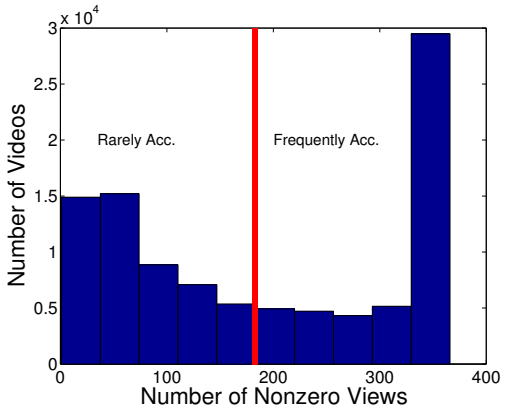


Fig. 1: Histogram of the number of days that have nonzero views.

are accessed on less than half of the days in the year (rarely-accessed). Figure 2 shows some examples illustrating the difference between frequently-accessed and rarely-accessed videos.

III. Methods

In this section, we briefly introduce the methods used for characterization and forecasting video time series.

A. Singular Value Decomposition (SVD)

For any $m \times n$ real matrix X, there exists a factorization of the following form:

$$X = U\Sigma V^T = \sum_{i=1}^{\min(m,n)} \sigma_i u_i v_i^T$$

where U and V are orthonormal matrices such that $UU^T = I$ and $VV^T = I$. Let u_i and v_i be the i^{th} columns of U and V respectively. Matrix Σ is a $m \times n$ diagonal matrix where each diagonal entry is a singular value, σ_i. The matrix Σ is arranged in such a way that $\sigma_i \geq \sigma_{i+1}$. This factorization is called the *Singular Value Decomposition (SVD)* of X.

One of the most popular applications of SVD is matrix approximation, i.e. approximating a matrix X with another matrix $\tilde{X}$ of lower rank r. To find a matrix $\tilde{X}$ with rank r that minimizes $||X - \tilde{X}||_F$, once can use the SVD of X as follows:[1]

$$\tilde{X} = U\tilde{\Sigma}V^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$

There are two pre-processing steps that may be applied on matrix X prior to SVD. The first one is mean centering, i.e. subtracting the column mean from each entry. The second is to normalize each entry by the l_2-norm of its column.

B. Autoregressive Moving Average Model

An Autoregressive Moving Average (ARMA) model is one of the most popular methods for modeling and predicting future values of a time series [1]. It consists of two parts: an Autoregressive (AR) model and a Moving Average (MA) model. Given a time series Y, an AR model of order p is defined as:

$$Y_t = \sum_{i=1}^{p} \alpha_i Y_{t-i} + \epsilon \tag{1}$$

where $\alpha_1, ..., \alpha_p$ are the parameters of the model and ϵ is a white noise error term. An MA model of order q is defined as follows:

$$Y_t = \epsilon_t + \sum_{j=1}^{q} \theta_j \epsilon_{t-j} \tag{2}$$

where $\theta_1, ..., \theta_q$ are the parameters of the model and $\epsilon_t, ..., \epsilon_1$ are again white noise error terms. Combining Equations (1) and (2), an ARMA model of order (p, q) is written as follows:

$$Y_t = \sum_{i=1}^{p} \alpha_i Y_{t-i} + \epsilon_t + \sum_{j=1}^{q} \theta_j \epsilon_{t-j} \tag{3}$$

The error terms, ϵ_t, are generally assumed to be Gaussian i.i.d. random variables with zero mean and constant variance.

IV. Frequently Accessed Videos

With these tools in hand, we can now describe our main results. As mentioned above, frequently-accessed videos are those that are continuously popular during the year, i.e. viewed almost every day. For this analysis, we concentrate on the most popular 1000 videos as measured by the total number of views.

A. Characterization

Our characterization focuses on (1) understanding common patterns in data, and (2) using that understanding as an aid to prediction.

We observe that there are temporal correlations in our frequently-accessed data set. By employing SVD, we decompose the time series into their main constituents. Let X be a 366×1000 matrix, where each column of X is a 366-day time series of a video. Prior to SVD, we mean center and normalize X as explained in Section III-A. Figure 3a demonstrates the magnitudes of the singular values of X. In this figure, it is seen that there is a knee around the 20^{th} singular value. To be more accurate, 88% of energy level is achieved by largest 20 singular values, where total energy is defined as a function of singular values, σ, of X as follows: $\sum_{i=1}^{366} \sigma_i^2$. This suggests that there is a considerable structure and only 20 principal components are enough to approximate our collection of videos.

The largest three principal components are presented in the top row of Figure 4. The first principal component shows a steady increase during the year. The second principal component shows increase until the middle of the year and then decrease. The third component shows two fluctuations during the year. One common behavior in these principal components is that they all show distinct 7-day fluctuations.

In the next section, we show that this compact representation helps efficiently predict the daily accesses that videos receive in the future.

[1] The Frobenius norm of an $m \times n$ matrix M is $||M||_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} M_{ij}^2}$.

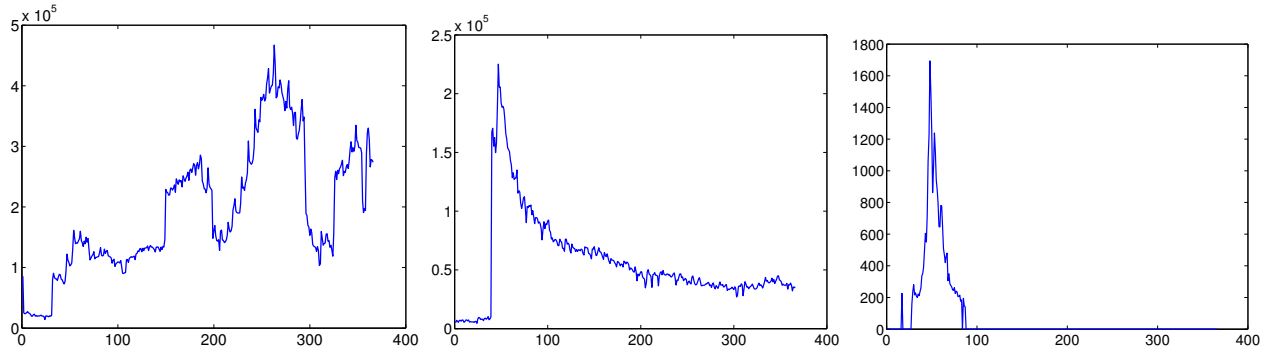
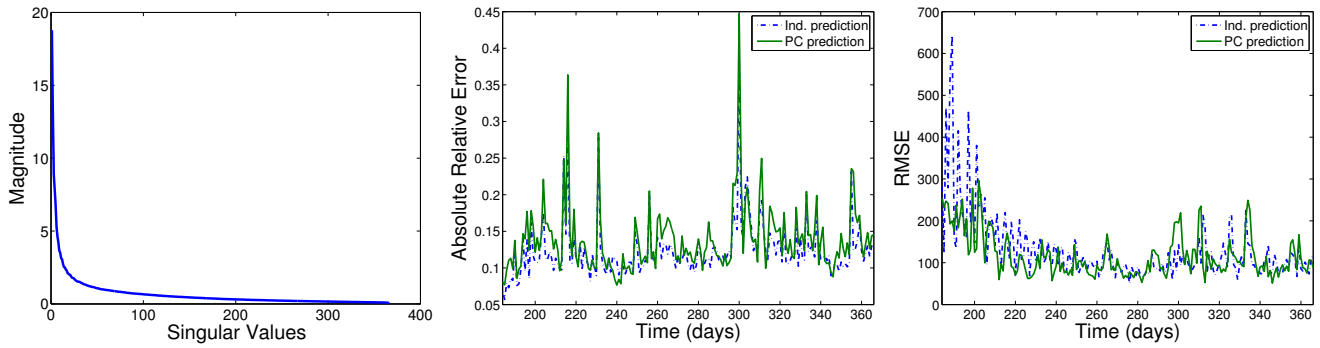


Fig. 2: Example time series of video accesses: frequently-accessed (the figures on the left and the center) and rarely-accessed (the figure on the right). The x axis is time (in days) and the y axis is the number of daily views.



(a) Singular Values of Most Popular 1000 Videos (b) Absolute Relative Error (c) RMSE

Fig. 3: Video access characterization and forecasting accuracy.

B. Forecasting

As described in Section III-B, one of the most popular techniques for modeling and predicting time series is ARMA modeling. To assess the utility of ARMA modeling for our data, we first apply this method on each video individually. To set the order of the model (p, q), we use one-day-ahead ARMA predictions with order values ranging from $(4, 4)$ to $(12, 12)$. We find that $(7, 7)$ is the smallest order that yields good results. This suggests that using observations of one week past is sufficient for modeling the behavior of the next day. In fact, this is understandable in light of the weekly fluctuations seen in our time series.

For each time series, we use the first half of the year as the training set for generating an ARMA model. Then for each of the remaining 183 days, we forecast one day ahead.[2]

To define an error metric, let X_{ij} be the true view count and $\hat{X}_{ij}$ be the predicted view count on day i for video j. Then, average absolute relative error is defined as $\frac{1}{N}\sum_{j=1}^{N}\frac{|\hat{X}_{ij}-X_{ij}|}{X_{ij}}$ and average root mean squared error (RMSE) is defined as $\frac{1}{N}\sqrt{\sum_{j=1}^{N}\left(\hat{X}_{ij}-X_{ij}\right)^2}$, where N is the number of videos.

We find that ARMA modeling works successfully for forecasting future daily accesses. The dashed line in Figure 3b and Figure 3c illustrate average absolute relative error, and average RMSE, respectively. Errors are averaged over all videos from day 184 to 366. The average absolute relative error is below 0.15 and RMSE is below 200 for most days.

While this method shows good accuracy, its computational cost is unfortunately high, and scales with the number of videos to forecast, because it requires generating a model for each video separately. Our strategy for making cost managable combines the two observations made so far: approximation via PCA and forecasting with ARMA models. Our approach is to apply ARMA modeling on the principal components of the data instead of the individual time series. In other words, instead of directly forecasting the individual time series, we forecast the principal components, an approach we call *PC forecasting*. Just as for individual time series forecasting, in PC forecasting we use the first 183 days as the training set to generate ARMA models with order $(7, 7)$ and predict one-day ahead for the rest of the year. The bottom row of Figure 4 shows the ARMA predictions of the first three principal components. As can be seen, ARMA $(7, 7)$ models can accurately forecast the principal components.

However, our main goal is predicting not the principal components but the original daily views. This requires transforming PC forecasts into the individual forecasts. Let $X_{1:t}$ be the rows of X from day 1 to day t and $\tilde{X}_{1:t}$ be the preprocessed form of $X_{1:t}$ prior to SVD (see Section III-A). $\tilde{X}_{1:t}$ is decomposed into its $U_{1:t}$, Σ', and V' after SVD is

[2] Note that forecasting more than one day ahead is possible, albeit with less accuracy. We omit this analysis for lack of space.

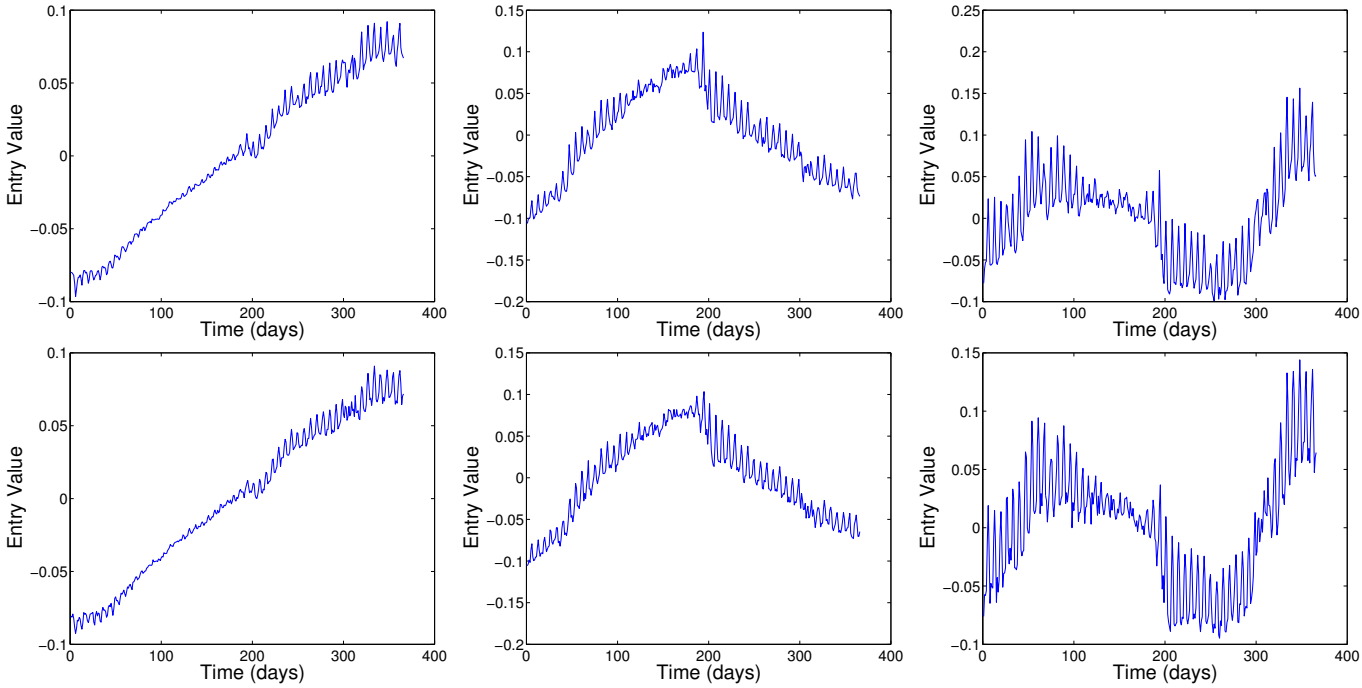


Fig. 4: The largest three principal components (upper) and their ARMA predictions (lower). The x axis is the time (in days) and the y axis is the magnitude of the principal component.

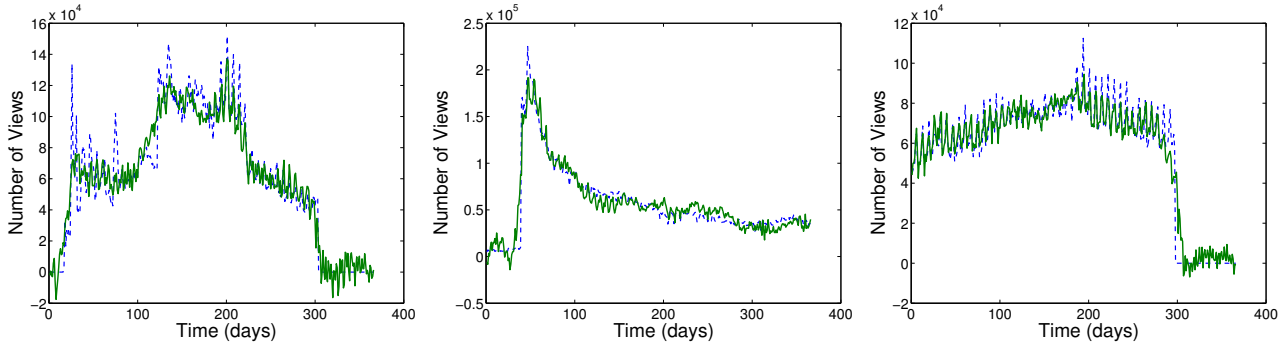


Fig. 5: Examples of one-day ahead PC forecasting. Dashed lines are actual time series and solid lines are PC forecasts.

applied. The columns of $U_{1:t}$ are the principal components of matrix $\tilde{X}_{1:t}$ and $\tilde{U}_{1:t}$ represents the first 20 columns of $U_{1:t}$. Initially, an ARMA model is generated for each column of $\tilde{U}_{1:183}$. Then, by using these models, on any day t, $\tilde{U}_{t+1}$ can be computed. At this point, we can approximate $\tilde{X}_{1:t+1}$ as the product of $\tilde{U}_{1:t+1}\Sigma' V'^T$. The last step is to reverse the pre-processing prior to SVD, by converting $\tilde{X}_{1:t+1}$ to $X_{1:t+1}$.

Figure 5 shows some examples of time series and their PC forecasts. It shows that PC forecasting can be very successful in predicting the next day's accesses. To obtain a sense of overall error, Figures 3b and 3c compare the performance of PC forecasting and individual forecasting. In both figures, the dashed line represents the day by day error in individual forecasting and the solid line represents error in the PC forecasting. The x-axis starts from day 184, since we start forecasting on day 183. In Figure 3b, for both individual and PC forecasting, the absolute relative error is quite low. The mean absolute relative error is around 0.12 for individual forecasting and 0.14 for PC forecasting. Figure 3c shows the RMSE on each day. The mean RMSE is about 130 for individual forecasting and 117 for PC forecasting. These values are low given that the daily views of the videos range between the scale of 10^4 - 10^7.

While the increase in error due to PC forecasting is small, the improvement in scalability is large. PC forecasting requires training only 20 ARMA models, a number that is not expected to change significantly as the number of videos modeled grows. Even for only 1000 videos, this is a considerable saving in running time. For example, on a four processor 2.66GHz Intel with 4GB of RAM, running 64-bit Linux, individual forecasting takes 844 seconds, whereas PC forecasting takes 150 seconds to finish. Thus, for our set of 1000 videos, PC

forecasting is about 5.5 times faster than individual forecasting.

In sum, we see that exploiting the structure inherent in the data means that, instead of constructing thousands of ARMA models, one only needs to construct a small number of models. This provides a significant improvement in scalability, with very little penalty in accuracy.

V. Related Work

Our works relates to a broad spectrum of topics, from workload characterization to compact representation of large data streams. In this section, we briefly mention related work on these topics.

A number of studies have examined the characteristics of user-generated video sharing systems. Among these studies, some specifically focus on YouTube [2], [3], [4], [6]. In [2], Cha et al. analyze the popularity distribution of YouTube videos and how users' requests are distributed across popular and unpopular videos. They also analyze the popularity evolution of videos, i.e., the change in popularity as the videos get older. They show that an unpopular video is unlikely to get popular as it ages. Based on these observations, they suggest that future popularity of videos could be predicted but do not provide a way doing so. In [3], Cheng et al. study statistical properties of YouTube videos such as the distribution across different video categories (e.g. music, sports etc.), video lengths, active life span of videos, and growth trend in uploading new videos. One finding in that paper related to ours is that most videos have a short active life span. This is consistent with the fact that almost 50% of the videos in our dataset are in the rarely accessed category. In [6], Gill et al. characterize YouTube usage from an edge network perspective by studying characteristics such as file size, video durations, video bit rates and usage patterns within a campus network.

Our work differs from these works in several aspects. First, we use a complete and global dataset observed at YouTube servers. Our collection of videos is not biased by the recommendation system of YouTube, a specific group of users, or video categories (e.g. entertainment or sports). Second, our work does not depend on meta information, such as how long it has been since the video was uploaded, its rankings, etc. Most importantly, these previous studies do not propose a framework that can be used for quantitative forecasting.

There are also studies that focus on the social networking aspects of YouTube. In [4] Cheng et al. propose a peer-to-peer short video sharing framework that leverages social networks. In [5], Crane et al. investigate how a social system responds to bursts of exogenous and endogenous activity by using the time series of daily views of YouTube videos, and they find four different shapes for bursts. Their models are relatively simple compared to our clustering-based results, which reveal more complex shapes.

Finally, another set of related work concerns representing large data streams with smaller-sized approximations. This is a well studied topic with wide application in the field of data mining. Two typical examples are [10] in which Korn et al. use SVD for data compression of large data sets, and [11] in which Papadimitriou et al. summarize the key trends in data streams by extracting their principal components. In both cases, however, these results are not used as tool for forecasting, but rather as a general form of data compression.

VI. Conclusion

In this paper, we have analyzed a large dataset consisting of daily access counts of hundreds of thousands of YouTube videos. We find that there are two types of videos: those showing rapid changes in popularity, and those that are consistently popular over long time periods. Our study shows that, for consistently popular videos, there is a relatively simple model that can describe the daily access patterns and this simple model can be effectively used to predict the number of accesses that a video will have in the near future. We show that for frequently-accessed videos, daily access patterns can be extracted via principal component analysis, and used efficiently for forecasting via autoregressive models. Keeping in mind the importance of video-sharing as a traffic driver and workload type in today's Internet, our results represent a useful step towards efficient and effective forecasting for video-sharing sites.

VII. Acknowledgments

This work has been partially supported by National Science Foundation awards: CNS-0905565, CNS-1018266, CNS-1012910, CNS-0963974, CCF-0820138, and CSR-0720604. The authors are grateful to Google Inc., and in particular Leonidas Kontothanassis, for providing the data used in this study.

References

[1] P. J. Brockwell and R. A. Davis. *Time Series: Theory and Methods*. Springer Series in Statistics. Springer-Verlag, second edition, 1991.

[2] M. Cha, H. Kwak, P. Rodriguez, Y. yeol Ahn, and S. Moon. I tube, you tube, everybody tubes: Analyzing the worlds largest user generated content video system. In *In Proceedings of the 5th ACM/USENIX Internet Measurement Conference (IMC07)*, 2007.

[3] X. Cheng, C. Dale, and J. Liu. Understanding the characteristics of internet short video sharing: Youtube as a case study. *CoRR*, abs/0707.3670, 2007.

[4] X. Cheng and J. Liu. Nettube: Exploring social networks for peer-to-peer short video sharing, 2009.

[5] R. Crane and D. Sornette. Robust dynamic classes revealed by measuring the response function of a social system, 2008.

[6] P. Gill, M. Arlitt, Z. Li, and A. Mahanti. Youtube traffic characterization: a view from the edge. In *IMC*, pages 15–28, New York, NY, USA, 2007.

[7] G. Gürsun, M. Crovella, and I. Matta. Describing and forecasting video access patterns. Technical Report BUS-TR-2010-037, CS Department, Boston University, November 2010.

[8] http://asert.arbornetworks.com/2010/03/how-big-is google/.

[9] http://www.youtube.com/.

[10] F. Korn, H. V. Jagadish, and C. Faloutsos. Efficiently supporting ad hoc queries in large datasets of time sequences. In *SIGMOD '97: Proceedings of the 1997 ACM SIGMOD international conference on Management of data*, pages 289–300, New York, NY, USA, 1997. ACM.

[11] S. Papadimitriou, J. Sun, and C. Faloutsos. Streaming pattern discovery in multiple time-series. In *VLDB '05: Proceedings of the 31st international conference on Very large data bases*, pages 697–708. VLDB Endowment, 2005.